Azwad Shameem

CS 301 – Numerical Issues in Scientific Programming

Final Report

**Analysis Of Root Finding Methods**

The idea is basically of root finding methods started with the belief that any polynomial can be factored as **f(x) = π (x-ai)**, where a1….,n are the roots of the polynomial. Unfortunately, this belief is not always true, the Abel-Ruffini theorem, also known as Abel’s impossibility theorem, states that there are no solutions to general polynomial equations of degree five or higher with arbitrary coefficients. Furthermore, the Galois theory is another theory that supports the impossibility of a general equation for polynomials with a degree of five or higher. The Galois theory states the simplest non-solvable fifth degree polynomial example, x5 - x - 1 = 0, that cannot be solved in radicals. The same however, does not apply for degrees lower than five because there is the quadratic formula, the cubic formula and the quartic formula to find the roots for degrees two, three and four respectively. This creates a predicament because there is no general equation capable of solving for the roots of equations with the degree of 5 or above. In essence, Abel’s impossibility theorem and Galois’s theorem combined makes it seems highly convincing that polynomials with a degree of five or higher will not have a general formula to solve for their roots.

In order to combat the issue that no general formula to solve for the roots of fifth degree polynomials several algorithms were made to approximate the. A multitude of such algorithms were made in order to approximate the root and they all have differentiating reliability, speed, accuracy. In addition, each individual methods also have their own requirements in order to fully utilize their potential, take Newton’s root finding method for example, it requires the user to find the derivative of the function and use the derivative to then approximate the root. Another example, of individual algorithm requirements are that some methods require could either a single approximation, a double approximation, a triple approximation or even more approximations in order to use that particular method. Some algorithms even require the predication to either be near the root or two predictions that contain the root in the values in between the predictions.

It is of most importance to know which one of these methods’ advantages best suit itself for which tasks because these are currently the only few ways in order to solve for roots that are for polynomials with degree five or above. Therefore, to gauge each methodologies’ abilities they will be tested on three reliability tests and speed tests.

In order to accurately compare the different mythologies, we will first compare, them in the order of their input requirements. The reliability test will utilize equations

x5 - x – 1, 2x6 - 2x2 - 2, 5x7 - 4x6 - 4x – 9 and test predictions that are at least 5, 50 and 100 whole digits away from the root respectively. The speed test will utilize equations

x5 – x4 - 5x - 5, x6 – x4 - 5x - 5, 9x7 - 8x5 - 3x – 1 and test predictions that are at least 1, 2, 3 units away respectively. Lastly, the constants of the all the mythologies will be a tolerance of 1e-15 and a maximum number of iterations set to 1 million.

The Fixed-Point Iteration is one of the simplest methodologies known to solve for the root. The biggest advantage of the Fixed-Point Iteration method is its ease of implementation and simplistic, the method quite literally entails the use of one prediction and then repeated plugging the prediction into the equation and setting result of the evaluation of the equation at the prediction as the new prediction until it is not within tolerance levels. Unfortunately, the simplicity of the Fixed-Point Iteration method Is a double-edged sword because while it is easy to create it is very unreliable. The Fixed-Point Iteration’s reliability test results reflect its unreliability. Fixed-Point Iteration took the maximum number of 1 million iterations and still failed to get the root; in fact, its progression path didn’t show any progress towards the root either. Reliability tests two and three showed slight progress, it did not even reach the root and actually went so off from course that it would cause a stack overflow error if it was allowed more than three iterations. The three speed tests were similarly grim for the Fixed-Point Iteration because it was not able to find the root for any test and it caused an overflow error over 3 iterations. Clearly, the Fixed-Point Iteration is very unreliable and only suitable for a very limited section of equations if any at all. Since the Fixed-Point Iteration was not able to predict the root in any test, it is not possible to state or compare the number of iterations, error from root, time taken or even the path taken. Also, it is not possible to compare the Fixed-Point Iteration method because they are not accurate representations because of the overflow error that is frequent with this method for higher iterations caused the method to be halted before the maximum number of iterations. In conclusion, the Fixed-Point Iteration was an extremely disappointing display because it was never able to predict the root correctly in any test.

The next method is Steffensen’s method, this method requires one root just like the Fixed-Point Iteration method. The advantages of the Steffensen’s method are that it is by applying a modification of Aitken’s delta squared method to a linearly convergent sequence obtained from fixed-point iteration it can accelerate the convergence to a quadratic. Similarly, to Fixed Point Iteration, Steffensen’s method also failed to predict the root on all the reliability tests. The reliability tests show that Steffensen’s method is far from a reliable method and that there was a sizeable error from the actual root for all of the reliability tests. In addition, Steffensen’s method took 1 million iterations each time and had the highest amount of time required because it used all the iterations and still failed to find a root within the required tolerance. Surprisingly, Steffensen’s method was able to predict the root for the first speed test and have the smallest error for speed test 1, even though it took the most iterations and time for the first speed test. However, Steffensen’s method did not fare well with any of the speed tests afterward and its error from the root, number of iterations and time taken showed as such. These pieces of information show an interesting result from Steffensen’s method because even though it has only one prediction it was able find the root during the first speed test because the prediction was close enough to the root. This makes Steffensen’s method seem like it can be a reliable method if the prediction not too far away from the actual root because it was able to discern the actual root when it was around 1 unit away even though it may require more iterations and time than other methods. Therefore, this makes Steffensen’s method a somewhat reliable method if the user has a very good prediction of the root. Considering the fact that Steffensen’s only requires a single prediction it outweighs the negatives that it may be a slower than other methods. In essence, Steffensen’s method is an option to think about because if the prediction is close enough it can find the root with just one prediction.

The next method on line is a very unique method because it requires the derivative of the equation, this method is called the Newton’s method. The Newton’s method is a very powerful method because by utilizing the derivative and one prediction it can be exceptionally fast and reliable at times. The derivative is a key factor for the Newton’s method which is why this method is only recommended when finding the derivative is very easy because sometimes finding the derivative is harder than finding the root. Newton’s method uses the initial prediction and the approximation of the x-intercept of the tangent line to the graph of the equation at the initial prediction and the value of the equation at the initial prediction. Newton’s method successfully predicted the root for all of the reliability tests even though the prediction were as far as at least one hundred units away from the actual root. In terms of iterations, time Newton’s method was not the best and sometimes took the most time and iterations of the methods that successfully predicted the root. In fact, Newton’s method was actually the method with the most error out of all the equations that successfully predicted the root in all of the reliability tests. During the speed tests on the Newton’s method, it is clearly visible that the Newton’s method’s speed is heavily impacted by the closeness of the prediction to the root because when the predictions were 1 or 2 units farther away it took more iterations and time than some of the other methods. The error from the actual root had the same result as the reliability tests because it seemed like Newton’s method usually had the highest error of the methods that found the root successfully for all the tests even though the tolerance level is the same. In spite of all of the disadvantages of Newton’s method, the result of Newton’s method is very good because none of the one prediction methods used were not able to predict the root successfully predict the root on every test like the Newtons method. Therefore, the Newton’s method is extremely reliable and can be fast which makes the Newton’s method a go to method for many users if the derivative of the function is easy to find.

The single prediction methods represented are Fixed-Point Iteration, Steffensen’s method and Newton’s method. Newton’s method is the best method out of the single prediction methods, but it does require the derivative. The best method for only a single prediction without any other requirements is the Steffensen’s method even though it is not very reliable. It is important to understand that the single prediction methods are very limited because they have to work really hard to find the root since they are given the least information.

The first double prediction method is the Bisection method. The Bisection method is known for its reliability and guarantee to find the root as long as its requirements are satisfied. The requirements of the Bisection method to guarantee that the root will be found is that the two predictions should be represented as variable ‘A’ equals a number and variable ‘B’ is equal to a number that’s negative and that the range between variable ‘A’ and variable ‘B’ should contain the root. In addition, the Bisection method requires an adequate number of iterations to find the root depending on the tolerance. The Bisection method follows through with expectations on the reliability tests and speed tests because it was able to predict the root for every single test. One of the biggest advantages for the Bisection method beside guarantee to find the root is that the Bisection method is very precise in the number of iterations and time it takes for any equation. For both reliability and speed tests the Bisection method took about fifty-four iterations and around one tenth of a millisecond of time for easier and more complex equations. This shows that the Bisection method is very precise and is an extremely stable method, which makes it a very important method for users who prefer accountability over cutting edge speed from the method. Essentially, since Bisection method takes around the same iterations no matter the equation at the same tolerance, it makes the Bisection method a method for people to value safety and also for people who do not want to worry about errors when trying to find the root. In addition, the Bisection method usually has one of the lowest errors from the actual root even though the tolerance is the same for every method. As a result, the Bisection Method is probably the one of the most reliable methods and fits users who want to use a method that without will not make them worry about reliability or speed.

The second double prediction method is called the Secant method. The Secant method uses a succession of root predictions through secant lines to better approximate the root of a function. This methodology of utilizing secant lines has its advantages and its disadvantages